

09/893,809

MS174294.01 / MSFTP244US

REMARKS

Claims 1-27 are currently pending in the subject application and are presently under consideration. Claims 1, 6, 8, 9, 19, 23, and 26 have been amended as shown on pages 2-7 of this Reply. Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I. Rejection of Claims 9-18 Under 35 U.S.C. §101

Claims 9-18 stand rejected under 35 U.S.C. §101 because they are directed to non-statutory subject matter. Per the Examiner's suggestion, claim 9 has been amended to recite "computer implemented" method. Accordingly, this rejection should be withdrawn.

II. Rejection of Claims 23-25 Under 35 U.S.C. §101

Claims 23-25 stand rejected under 35 U.S.C. §101 because the Examiner contends they are directed to non-statutory subject matter. Withdrawal of this rejection is requested for at least the following reasons. The claims are properly directed toward statutory subject matter.

Section 271(f) refers to "components of a patented invention."... Title 35, section 101, explains that an invention includes "any new and useful process, machine, manufacture or composition of matter."... Without question, *software code alone qualifies as an invention eligible for patenting under these categories*, at least as processes. This statutory language did not limit section 271(f) to patented "machines" or patented "*physical structures*." Rather every form of invention eligible for patenting falls within the protection of section 271(f). By the same token, the statute did not limit section 271(f) to "machine" components or "*structural or physical*" components. *Rather every component of every form of invention deserves the protection of section 271(f).* *Eolas Techs., Inc. v. Microsoft Corp.*, 399 F.3d 1325, 1338-39 (Fed. Cir. 2005) (emphasis added).

09/893,809

MS174294.01 / MSFTP244US

At pages 2-3 of the Office Action (dated June 13, 2005), the Examiner incorrectly contends that a "data packet" is nonstatutory because it is incapable of being touched or perceived. The CAFC has recently held that software code alone (e.g., a data packet) qualifies as an invention. The statutory language does not limit patentability to physical structures (e.g., things that can be touched or perceived). *See Eolas Techs., Inc. v. Microsoft*, at *id.* Accordingly, this rejection should be withdrawn.

III. Rejection of Claims 1-7, 26 and 27 Under 35 U.S.C. §101

Claims 1-7 and 26-27 stand rejected under 35 U.S.C. §101 because the Examiner contends they are directed to non-statutory subject matter. Withdrawal of this rejection is respectfully requested for at least the following reasons. The claims are properly directed toward statutory subject matter.

The Examiner states at page 3 of the Office Action that the components of the systems "*appear to be software modules, which are not tangible. Therefore, [the claims are] non-statutory because [they] recite a system that comprises non-tangible embodiments.*" Pursuant to *Eolas Techs., Inc. v. Microsoft*, neither systems nor components of systems need be structural or physical to be statutory subject matter. Accordingly, this rejection of independent claims 1 and 26 as well as all associated dependent claims should be withdrawn.

IV. Rejection of Claims 1-27 Under 35 U.S.C. §103(a)

Claims 1-27 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Kind (US Patent 6,415,434 B1) and further in view of Admitted Prior Art (APA). Withdrawal of this rejection is respectfully requested for at least the following reasons. Kind, alone and/or in combination with alleged APA, does not teach or suggest all limitations set forth in the subject claims.

09/893,809

MS174294.01 / MSFTP244US

To reject claims in an application under §103, an examiner must establish a *prima facie* case of obviousness. A *prima facie* case of obviousness is established by a showing of three basic criteria. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) ***must teach or suggest all the claim limitations***. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must be found in the prior art and not based on the Applicants' disclosure. See *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

Independent Claims 1 and 8

Applicants' claimed invention relates to a system that facilitates interactions between a first entity and a second entity, where the entities have a mismatched data type with at least one aspect in common. In particular, amended independent claim 1 (and similarly amended independent claim 8) recites, "***a data type identifier that identifies whether the first entity and the second entity have a mismatched resolvable data type; and a data type resolver that receives the mismatched resolvable data type from the data type identifier***". Kind fails to teach or suggest these novel aspects.

Rather, Kind relates to resolving overloaded class methods during runtime. Kind discloses a resolver (see FIG. 1, item 104) that receives a target method from a third party application (see col. 5, ll. 63-65), and retrieves pre-defined methods from an application programming interface (API). (See FIG. 1, item 128; col. 5, ll. 30-32). Kind fails to teach or suggest "***a data type identifier***" from which the resolver "***receives the mismatched resolvable data***". According to the cited reference, after the target method has been received, and the resolver retrieves the pre-defined methods from the API, the resolver chooses the method that exactly matches the target method. (See col. 3, ll. 35-37; ll. 51-53; col. 4, ll. 3-5). If there is no exactly matching method, the resolver (as opposed to a data type identifier) selects a best method that most closely matches the target method. (See col. 3, ll. 38-40; ll. 54-56; col. 4, ll. 5-8). Accordingly, Kind does

09/893,809

MS174294.01 / MSFTP244US

not teach or suggest a data type resolver that receives the mismatched resolvable data type *from the data type identifier*.

To further illustrate this distinction, Kind requires a User Interface (UI) Description File (see FIG. 3, item 314) to map the events to a callback function (see col. 6, ll. 21-25), and all such events are processed by the resolver irrespective of whether or not there is a mismatch and/or the event is resolvable. Hence, Kind cannot even detect there is a mismatch or if it is resolvable until the resolver first determines there are no exact matches. Therefore, the reference does not provide the flexibility of the subject invention, which need not invoke the resolver *unless and until* a resolvable mismatch has been identified. In Kind, the third party application does not have access to the class methods or the inheritance relationships (see FIG. 1) so it cannot identify if the data is mismatched and/or resolvable. Consequently, since the resolver is where the determination is made whether or not there is an exact match, the resolver does not receive mismatched resolvable data *from the data type identifier... that identifies* whether the first entity and the second entity have a mismatched resolvable data type. Accordingly, the Examiner has failed to make a *prima facie* case for obviousness because Kind does not disclose the limitations as recited the subject claims. Accordingly, withdrawal of this rejection is requested.

Independent Claims 8, 9, 19, and 26

Independent claim 19 (and similarly independent claims 8, 9 and 26) recites, “*creating an object of a third data type*, where the third data type comprises the at least one aspect common to the first data type and the second data type”. Kind does not teach or suggest these novel features.

Rather, Kind teaches the resolver receives a target method to compare with all methods of the target method class. If there is no exact match, then the resolver compiles a list of candidate methods, returning the best method or generating an error if there is no best method. (See FIG. 2, items 208-216). Therefore, Kind may compare aspects in common between the target method (*e.g.*, a first data type) with the candidate methods (*e.g.*, a second data type) in order to choose the best method, but Kind does not *create an object of a third data type*. Kind may further assign data type attributes of the target

09/893,809

MS174294.01 / MSFTP244US

method *via* inheritance or type conversion to cast the target method into the best candidate method (*see* col. 10, ll. 55-57); however, Kind returns the best candidate method, and the best candidate method is the second data type that pre-existed, not a newly created third data type. Kind does not contemplate creating an object of a third data type, but instead, can only convert a first data type into a second data type (*e.g.*, the best method data type). Accordingly, Kind does not teach or suggest *creating an object of a third data type*, where the third data type comprises the at least one aspect common to the first data type and the second data type. Thus, it is requested that this rejection be withdrawn.

Independent Claim 23

Applicants' claimed invention further relates to data types that are incrementally extensible. When data types are mismatched, a determination can be made concerning whether the proxy on the client side should load all or merely a subset of the server-side proxy, and each proxy and/or subset can be extensible as needed. (*See* pg. 23, ll. 15-17). In particular, amended independent claim 23 recites, "one or more first fields containing information concerning attributes associated with a first data type, where the first data type is *incrementally extensible* and the attributes are *loaded on an as-needed basis*". Kind does not teach or suggest these novel aspects.

Kind does not contemplate *incremental extensibility*, nor does the reference consider whether one data type, although not identical to a second data type, can still contain all the necessary items (*e.g.*, attributes, methods, interfaces) to support interaction with the second data type without loading the additional items into, for example, client memory (or loading only a subset of the items). Rather, Kind returns either an exactly matching method or a best method, neither of which is contemplated to be incrementally extensible. The reference is silent on whether or not the attributes of the returned method can be loaded on an as-needed basis. Accordingly, Kind fails to teach or suggest one or more first fields containing information concerning attributes associated with a the first data type, where the first data type is *incrementally extensible* and the attributes are *loaded on an as-needed basis*, and this rejection should be withdrawn.

09/893,809

MS174294.01 / MSFTP244US

The alleged APA, which recites at the indicated portions "*Conventionally, a data type mismatch prevents the client and the server from interacting on the mismatched data types. With the increasing number of user created data types, such mismatching has increased*", does not make up for the aforementioned deficiencies of Kind. Accordingly, for at least the foregoing reasons, this rejection of independent claims 1, 8, 9, 19, 23, and 26 as well as all claims that depend there from should be withdrawn.

CONCLUSION

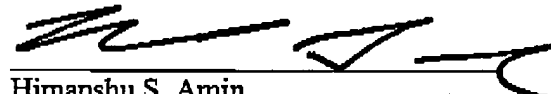
The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063.

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,

AMIN & TUROCY, LLP



Himanshu S. Amin
Reg. No. 40,894

AMIN & TUROCY, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731